BPbis Draft Evaluation

Scott Burleigh
Jet Propulsion Laboratory
California Institute of Technology

26 July 2019

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. © 2019 California Institute of Technology. Government sponsorship acknowledged.

BPbis Evaluation and Response

- draft-ietf-dtn-bpbis-13 has been evaluated by Magnus Westerlund, Area Director.
- The revised specification (to be posted on July 21) addresses the points in Magnus’s evaluation; also addresses Brian Sipos’s comments from May 16. Also renames “flow label” to “data label”.
- The following slides list the points raised in the evaluation and the proposed responses.

1. Section 4.1.5.1 and 10

As this document uses the "dtn" URI scheme and the "dtn" scheme was only provisional registered by RFC 5050 I think this document should formally register the DTN URI scheme and thus a new subsection in Section 10 is needed.

Yes. Section 10.3 has been added for this purpose.

Also, maybe similar effort to move "ipn" to permanent is needed?

By all means. But I think the first step is to get the “dtn” scheme provisionally registered.

2. Section 4.1.6 (1 of 2)

I think the format and definition of this field are insufficient. It is lacking a reference or a normative description of what "Unix Epoch Time" in an unsigned int format actually is. Is the intention here to simply measure the number of seconds since the start of year 2000? Which Unix epoch time is not due to leap seconds. And I become uncertain as UTC time scale is something else than Unix Epoch time at least when it comes to leap seconds handling. Due to that Unix Time have discontinuities in the time scale at leap seconds I would recommend strongly against using it. Select UTC or TAI depending on where you want the leap second handling pain to exist.

As we've discussed, some explanatory text is needed; it has been inserted into section 4.1.6.

2. Section 4.1.6 (2 of 2)

Also, you specified only CBOR unsinged integer which is a representation that can use 64-bit and thus not have the issues with 32-bit signed integer seconds epochs that regular unix time has, but that should probably be made explicit that it is expected to handle that. Even if the first wrap of this unsigned 32-bit format will not occur until 2136.

Noted in 4.1.6.

And please provide necessary normative references, not informative ones.

Some guidance needed here: which normative references are required?

3. Section 4.2.2

The Lifetime field appears problematic if the creating node doesn't have a reliable clock and uses timestamp 0 for all bundles. I think that special case needs to be discussed. Which it is later done, but there is missing reference to that.

Forward reference to 4.3.2 has been added in 4.2.2.

4. Section 4.2.2: Report-to EID

Is this field set by bundle creator and never changed? It is not clear and appear to have implications on how this field should be treated. Primarily considering integrity options over these fields.

> Text has been added in 4.2.2 to clarify that the primary block is immutable.

5. Section 4.3.3

Maximum Hop count upper limit? Can I insert a max UiNT64 here and expect that to be acceptable?

We left this question as explicitly “beyond the scope”, but we can assert a number. What should the maximum hop limit be?

6. Section 5.1 (1 of 3)

Under some circumstances, the requesting of status reports could result in an unacceptable increase in the bundle traffic in the network. For this reason, the generation of status reports MUST be disabled by default and enabled only when the risk of excessive network traffic is deemed acceptable.

I find the above paragraph rather unclear. First of all what are the circumstances. Secondly, shouldn't the type of status report requested be discussed. What I can see the different types results in at most 1, N-1 or N times the number of sent bundles with the status report set depending on type, where N is the number of nodes on the path including receiving endpoint.

Text acknowledging this has been added to 5.1.

6. Section 5.1 (2 of 3)

So are there any recommendations to when it might be acceptable to request status delivery. To me it appears the trace route like options, like forwarding status report should only be used on a small number of bundles sent for debugging or monitoring purposes.

That's the intent. A clarifying phrase has been added in 5.1.

Use of the bundle deletion status report appear to have other motivations for its use. I assume that this is fine for cases when it occurs for a small fraction of the bundles for some reasons, but when you have real failures on next hop links it appears that this may cause high volumes of deletion. Thus a rate limitation makes sense. I assume that this is one of the not specified reasons the below paragraph indicates:

6. Section 5.1 (2 of 3)

When the generation of status reports is enabled, the decision on whether or not to generate a requested status report is left to the discretion of the bundle protocol agent. Mechanisms that could assist in making such decisions, such as pre-placed agreements authorizing the generation of status reports under specified circumstances, are beyond the scope of this specification.

That's right. We believe that policy mechanisms to accomplish this are going to be needed, but we don't know what form they will take. Again, beyond the scope.

7. Section 5.3

Step 1: If the bundle's destination endpoint is an endpoint of which the node is a member, the bundle delivery procedure defined in Section 5.7 MUST be followed and for the purposes of all subsequent processing of this bundle at this node the node's membership in the bundle's destination endpoint SHALL be disavowed.

I don't understand why and what implications the disavowing has for a local bundle being dispatched from a member node part of the destination.

Some clarifying text has been added to 5.3.

8. Section 5.4

Step 3: If forwarding of the bundle is determined to be contraindicated for any of the reasons listed in Figure 4, then the Forwarding Contraindicated procedure defined in Section 5.4.1 MUST be followed; the remaining steps of Section 5 are skipped at this time.

Should that last Section 5 be Section 5.4?

Yes! Good catch – now fixed.

9. Section 6.1.1

What does Destination endpoint ID unintelligible actually mean?

Same with Block unintelligible.

> It means that the receiving bundle protocol agent cannot process the block for some reason: either the block is malformed or it is corrupted (CRC doesn’t match) or it is simply a type of extension block that the implementation does not recognize.

Are these the combination for corrupted blocks or EIDs? Are there a point of separating out the case where the CRC indicate a corruption?

> Yes. Added a new Step 3 in 5.6.

10. Section 7.2 (1 of 2)

So the service description appears very high level. How is the actual interface working when it comes to dealing with that there is either possibility to send, as well as rate control in the API. When can more data be accepted and when is the convergence layer not ready. Also don't the Bundle Agent need a signal when this side thinks it has delivered as a signal of when the forwarding has completed?

We specifically don’t want to constrain implementations of BP and convergence-layer adapters any more than necessary, as this has no impact on protocol interoperability. But yes, a “forwarding completed” signal is needed; inserted second bullet in list in 7.2.

10. Section 7.2 (2 of 2)

I was expecting this section to actually be explicit about what functionalities the Bundle Agent really need from the convergence layer.

> We really want to avoid over-constraining here. Implementation experience has shown that more extensive guidance is not needed.

11. Section 9

I think this section needs to be more explicit about mandating implementation support for BPSEC. The IETF do not publish a protocol today that doesn't have a mandatory to implement security solutions for the protocol's major properties. In this case communication security, i.e. confidentiality, integrity and authentication of the data communicated is very relevant. I have not yet read BPSEC so I may have additional concerns about that issues are not handled in the combined protocol. I hope the BPSEC has some discussion of the privacy properties of the protocol.

> Section 9 now states that inclusion of the Bundle Security Protocol in any Bundle Protocol implementation is REQUIRED.

12. Section 9

Additionally, convergence-layer protocols that ensure authenticity of communication between adjacent nodes in BP network topology SHOULD be used where available, to minimize the ability of unauthenticated nodes to introduce inauthentic traffic into the network.

In this context wouldn't it be reasonable to also recommend using encryption on the convergence layer to avoid eavsedropping on the part that is in clear and prevent traffic analysis by third parties?

Yes; added in 4th paragraph of section 9.

13. Section 9 (1 of 2)

Note that the generation of bundle status reports is disabled by default because malicious initiation of bundle status reporting could result in the transmission of extremely large numbers of bundle, effecting a denial of service attack.

I think there is a clear lack of mitigations proposed for this issue. As I mentioned in Issue 6 I think one both needs to consider amount of generated traffic versus utility for the sender. Also I will have to read BPSec to understand what integrity and source authentication there is of the request for status reports. Next is the issue of rate limiting and prioritization of status reports versus other bundles. Also due to the multi-hop store and forward nature of this protocol the actual bottle neck may only occur several hops towards the receiver. What can be said about dropping status reports versus other bundles when there is a resource contention, either in storage or in

13. Section 9 (2 of 2)

convergence layers capability of forwarding messages within time.

Yes. To address this, an additional reason code “Traffic pared” has been added to Figure 4. This authorizes the bundle protocol agent to drop status reports at its own discretion as described in Step 2 of 5.4.

14. Section 6

I fail to see any discussion of how the sender of a status report should set the lifetime and max hop. Can it actually take that information from the Bundle it is reporting on?

> There may be better ways to select values for these parameters, but none are known yet. The correct procedure may be wildly different in different kinds of delay-tolerant networks. We want to avoid over-constraining applications.

15. Section 10 (1 of 3)

I think this section needs to be clearer. Several improvements that can be made.

- I recommend individual sub-sections per registry operation
- Can you be clearer in references to point to specific sections of definitions that makes it simpler to find the relevant from the IANA registry?

Yes, done.

15. Section 10 (2 of 3)

More specific parts.

This document defines the following additional Bundle Protocol block types, for which values are to be assigned from the Bundle Administrative Record Types namespace [RFC6255]:

First of all according to the registry, this registry is actually created by RFC 7116.

RFC 7116 defines the Licklider Transmission Protocol. This sounds like an error in the registry.

15. Section 10 (3 of 3)

This and the observation that the things you attempt to register in this registry you are actual called Bundle Block Types in your document. Thus, I have to ask are you actually addressing the right registry, or is the issue that you actually need per version specific registries for example Bundle Block Types? If it is the later, then lets define new registries. Possibly there should be a new major page for BPv7. Not having read all the old documents, you likely have to consider which registries are version specific and which are not.

"Bundle Block Types" is correct. The block types defined for BPv7 are a superset of the block type types defined for RFC 5050.

16. Section 10

For the new registry, do you have any requirements for registrations that should be written out. And in addition any criteria you want the expert to consider when approving or rejecting registries? And is this registry version 7 specific or not?

No known requirements or considerations aside from the obvious ones (e.g., a reference document is required). RFC 5050 has no notion of URI scheme type numbers, so while the new registry is specific to version 7 it doesn’t supersede any other registry.

17. Section 4.1.1

The CRCs are lacking proper normative references. Needed for both 1 and 2. Note that you have [CRC] that is currently unused.

Right, added. (Brian Sipos commented on this as well.)

18. Section 11.2

[BPSEC] is a normative reference.

[RFC6255] is normatively referenced for their registration procedures.

Changed [BPSEC] to normative. Changed [RFC6255] to normative also, but since this is an informative RFC I worry that the nits filter won't like it.

19. Section 13, Section 4.2.2

This Section 13 item was something I wondered over:

Restructure primary block, making it immutable. Add optional CRC.

Did I simply miss where that is said in the context of Section 4.2.2?

Added text about primary block immutability in 4.2.2.

20. Optional CRCs (1 of 2)

Why are the primary block CRC optional? I assume the intention with it is to do a verification that the primarily block with the addressing information hasn't become corrupted in the transmission between nodes, similar to the checksums we have in other protocols. Under which conditions will not using it be a reasonable idea? Are you expecting other mechanisms to cover for it, or are you reasoning that having corrupted bundles being delivered to the wrong places is not an issue? As the primarily block is the main addressing part, I would think the considerations that went into discussion of the (almost) always usage of the UDP checksum for IPv6 applies here. Also the implications for corruptions and cost in some DTNs for stray data appears quite high.

> The thinking has been that you should virtually always have a bpsec BIB attached to the primary block, which not only authenticates the block but

20. Optional CRCs (2 of 2)

also protects the integrity of the block. In that scenario, a CRC on the primary block would not add much value.

BIBs can also be attached to other blocks, but in some cases a BIB might be overkill and a CRC is all that’s needed. So the CRCs are always an option.